

Modeling and Analysis of the Thermal Properties Exhibited by Cyber Physical Data Centers

Saif U. R. Malik, *Member, IEEE*, Kashif Bilal, *Student Member, IEEE*,
Samee U. Khan, *Senior Member, IEEE*, Bharadwaj Veeravalli, *Senior Member, IEEE*,
Keqin Li, *Fellow, IEEE*, Albert Y. Zomaya, *Fellow, IEEE*

Abstract— The Data Center (DC) contributes towards the prevalent application and adoption of the cloud by providing architectural and operational foundation. To perform sustainable computation and storage the DC is equipped with tens of thousands of servers, if not more. It is worth noting that, the operational cost of the DC is being dominated by the cost spent on energy consumption. In this paper, we model DC as a Cyber Physical System (CPS) to capture the thermal properties exhibited by the DC. All software aspects, such as scheduling, load balancing, and all the computations performed by the devices are considered as the “Cyber” component. The supported infrastructure, such as servers and switches are modeled as the “Physical” component of the CPS. We perform a detail modeling of the thermal characteristics displayed by the major components of the CPS. Moreover, we propose a Thermal Aware Control Strategy (TACS) that uses High Level Centralized Controller (HLCC) and Low Level Centralized Controller (LLCC) to manage and control the thermal status of the cyber components at different levels. Our proposed strategy is testified and demonstrated by executing on a real DC workload and comparing it to three existing strategies, one classical and two thermal aware strategies. Further, we also perform formal modeling, analysis, and verification of the strategies using High Level Petri Nets (HLPN), Z language, Satisfiability Modulo Theories Library (SMT-Lib), and Z3 solver.

Index Terms— Cyber Physical Systems, Cloud Computing, Data Center, Formal Methods, Modeling, Verification

1 INTRODUCTION

The DC hosts a large number of servers to improve the services for high performance computing application [1, 2].

Because of the high energy requirements of the computing and cooling devices, the DCs energy consumption can cost millions of dollars. The DC run-time cost is dominated by the cost spent on the energy consumption of computing and cooling technologies. Based on the energy consumption of a Google DC, a report suggested that Google was possibly running about 900,000 servers in 2010 [4]. The computational and operating margins of DCs depend highly on the provision of the QoS. Higher QoS attribute levels lead to higher rates that in turn lead to higher computations. To deliver the specified level of performance, the number of computational devices put in use at all levels of DC has significantly increased. As a result, the rate at which the heat is emitted by the devices has also increased. The cost to stabilize the temperature in the DC has drastically increased and become almost equal to the cost of operating computational systems. The increasing cost of energy consumption calls for new strategies to improve the energy efficiency in DCs. Several strategies have been proposed, such as [6, 7, and 8] for efficient energy consumption in the DC. In this paper, we

model DC as a Cyber Physical System (CPS) to capture the dynamics and evolution of the thermal properties presented by the DC. The phenomena of increase in the temperature of servers as a result of task allocations and the ambient effect of such increase in the temperature that affect other servers, is termed as the thermal dynamics of DC.

The software aspects, such as scheduling and computations, performed by the devices are modeled as the “Cyber” portion and the devices, such as servers and switches, are modeled as the “Physical” portion of the CPS. Several studies are available that model DC as a CPS to achieve energy efficiency, such as [9, 10, and 28]. The models proposed in the literature are abstract in the sense that they lack detailed analysis of the DC and hence it becomes difficult to exactly understand the process of heat distribution, both from software and infrastructure perspective. Thus, in this paper, we provide a detailed modeling and formulation of the cyber and physical infrastructure, including the heat dissipation of individual components, the heat distribution, and recirculation among the physical portion of the CPS.

The physical infrastructure of the DC follows a hierarchical model (as shown in Fig. 2), where the computing resources reside at the lowest layer. The network infrastructure can be considered as a multilayer graph, where the servers and switches are vertices and interconnection amongst them are the edges. The servers, access switches, and aggregate switches are assembled in modules (referred as pod) and are arranged in three layers, namely: (a) access, (b) aggregate, and (c) server layer. We perform a thorough analysis and modeling of the thermal subtleties involved at each layer. In doing so, we model heat dissipation of servers, switches (access layer, aggregate layer, and core layer), and the aggregate impact of each component on the overall infrastructure.

- S. U. R. Malik, K. Bilal, are with Department of Computer Science, COMSATS Institute of IT, Pakistan. Email: saif_ur_rehman@comsats.edu.pk, kashifbilal@ciit.net.pk.
- S. U. Khan is with the Department of Electrical and Computer Engineering, North Dakota, State University, 1411 Centennial Blvd, Fargo ND 58105-5285. Email:samee.khan@ndsu.edu.
- B. Veeravalli is with Department of Electrical and Computer Engineering at the National University of Singapore 117576. Email: elebv@nus.edu.sg.
- K. Li is with Department of Computer Science, State University of New York at New Paltz, NY, 12561, USA. Email: lik@newpaltz.edu.
- A. Y. Zomaya is with School of IT, University of Sydney, Sydney, NSW 2006, Australia. Email: albert.zomaya@sydney.edu.au.

Contributions: By exploiting the thermal behavior of discrete elements, we propose a Thermal Aware Control Strategy (TACS) that uses High Level Centralized Controller (HLCC) and Low Level Centralized Controller (LLCC) to manage and control the thermal status of CPS at different levels, such as: **(a)** low (server) level, **(b)** high (access, aggregate, and core switch) level, **(c)** intra-pod level, and **(d)** inter-pod level. The complete details of all levels and controllers will be discussed in later sections. We perform the simulation of our proposed strategy on a real data center workloads, obtained from Center of Computational Research, State University New York at Buffalo. The traces have more than 22,000 jobs and the records are of one month time. Moreover, we perform a comparative analysis of our proposed strategy with one classical scheduling approach and two thermal aware approaches, namely: **(a)** First Come First Serve (FCFS), **(b)** Genetic Algorithm based thermal aware scheduling [3], and **(c)** Thermal Aware Scheduling Algorithm (TASA) [18].

In this study, we also made an effort to diminish the level of abstraction through detailed modeling and formal analysis of the CPS. We use High-Level Petri Nets (HLPN) and Z language for the modeling and analysis of the systems. The HLPN are used to: **(a)** simulate and **(b)** provide mathematical representation, and **(c)** analyze the behavior and structural properties of the system. Moreover, we performed the verification of the models using Satisfiability Modulo Theories Library (SMT-Lib) and Z3 solver. We performed the automated verification of the model by following Bounded Model Checking technique using SMT-lib and Z3 solver. To verify using SMT, the Petri net model is first translated into SMT along with the specified properties. Then, Z3 solver is used to check whether the model satisfies the properties or not. The contributions of the paper are as follows:

- formulating the thermal properties of major component involved in CPS, the effect of cyber activities on the physical properties of the DC, and vice versa;
- proposing a Thermal Aware Control Strategy (TACS) that uses HLCC and LLCC to manage, control, and coordinate between the cyber and physical portion to maintain unified thermal threshold range;
- conducting simulation and comparison of proposed strategy on a real data center workload and;
- modeling and analyzing the CPS in HLPN, and the verification of the model using SMT-Lib and Z3 Solver.

The rest of the paper is organized as follows: Section 2 will review some of the related work done in the domain of thermal management and CPS modeling of DCs; preliminaries tools and technologies used in the paper will be presented in Section 3; modeling of thermal properties exhibit by cyber physical DC is performed in Section 4; the proposed control strategies and controllers are described in Section 5; modeling, analysis, and verification of the controllers and strategies are discussed in Section 6; the comparison results of our strategy with Genetic Algorithm (GA) based approach are demonstrated in Section 7; and Section 8 concludes the paper followed by the references and bibliographies of the authors.

2 RELATED WORK

The paradigm shift has occurred in the DCs, where the cost of IT equipment or hardware is no longer the major portion of the overall cost, instead the cost of power and cooling infrastructure has crept in to be the primary cost driver. Thermal

imbalance can cause a hurdle towards achieving an efficient operational DC. The presence of the hotspots creates a risk of redlining servers that can cause them to fail prematurely. The power consumption and thermal properties of the devices are directly proportional to each other. Therefore, in this section we will discuss both power and thermal strategies. Several strategies have been proposed to balance the tradeoff between the power, cooling, and performance. There are multiple ways to control the power consumption and thermal properties of the servers, such as through active management of workload hosted on the servers by using admission control strategies, load balancing, and workload migration. The power consumption of the servers can also be tuned through physical control, such as Dynamic Voltage and Frequency Scaling (DVFS) and on-off state control [12]. The DVFS has already been implemented in the operating systems, where the CPU utilization drives DVFS controller to adopt the power consumption with the changing workload, such as in [15]. A technique to control the workload execution on the processor and the power consumption, given some constrained on the temperature of the chip, is proposed in [16]. Moore *et al.* [7] proposed a temperature aware workload placement approach in DC. The aforesaid approach is based on thermodynamics formulation, power, and thermal profiles of the servers. However, precise measurement of the profiles for such a large number and types of jobs is complicated. Moreover, the thermal and power models are not accurate for DC. In another approach [19], modeling a thermal topology of DC is discussed that can lead to more efficient workload placement. However, preserving the safe temperature and migration of the resources are not discussed. A DC environmental control system is proposed in [21] that uses a distributed sensors network to manipulate CRAC units. However, the discussion in [21] is concentrated only on the CRAC and did not consider the servers. In [10], the authors have modeled DC as a CPS and proposed a control strategy to optimize the tradeoff between the quality of computational and energy cost. However, the heat recirculation and its effect on the other neighboring nodes are not discussed.

In [22], the authors have proposed an analytical transient heat transfer model as a replacement of CFD simulations, to speed up the evaluation and decision making process in initial designing and modifying the configurations of the data center. The CFD simulations take considerable amount of time and such long stretch of simulation time is not suitable for online model-based decision making. To solve the aforesaid problem, the authors in [7] have proposed a transient heat transfer model that takes a small fraction of CFD run time and has the ability to introduce logic and triggers, which are hard to implement in CFD. In our paper, we model the heat dissipation of the discrete elements of cyber physical DC, such as servers and switches, as a function of power consumed by the devices when the processing is being performed. Once the thermal values of the devices are computed, we exploit those values to perform the task migrations and traffic redirection to avoid hotspots and maintain thermal balance within the DC. Moreover, the thermal values are further used to compute the ambient effect of a server in a close proximity by using thermodynamic concepts. Furthermore, the thermal effect of allocating task to a server and other neighboring servers is also analyzed using the aforesaid thermal values. To model the transmission of heat and its effect on other servers, we used thermodynamic concepts. The thermal analysis is then used to propose a thermal aware control strategy that maintains thermal uniformity within the pods of DC.

3 PRELIMINARIES

In this section we will discuss some of the tools and technologies used in this work that will help the reader to understand the paper easily.

3.1 High-Level Petri Nets (HLPN)

Petri nets are graphical and mathematical modeling tool that is applicable to many systems characterized as being concurrent, asynchronous, distributed, parallel, non-deterministic, or stochastic [24]. In this paper we have used a variant of classical Petri Net model, namely, High-Level Petri Nets (HLPN) (as shown in Fig. 1).

Definition 1 (HLPN) [24]. A HLPN is a 7-tuple $N = (P, T, F, \varphi, R, L, M_0)$ where: P is a set of finite places, T is a set of finite transitions such that $P \cap T = \emptyset$, F is a flow relation such that $F \subseteq (P \times T) \cup (T \times P)$, φ is a mapping function that maps P to data types such that $\varphi: P \rightarrow \text{Data Type}$, R define rules that map T to predicate logic formulas such that $R: T \rightarrow \text{Formula}$, L is a label that maps F to labels such that $L: F \rightarrow \text{Label}$, and M_0 is the initial marking where $M: P \rightarrow \text{Tokens}$.

3.2 SMT-Lib and Z3 Solver

Satisfiability Modulo Theories (SMT) is an area of automated deduction for checking the satisfiability of formulas over some theories of interest and has the roots from Boolean Satisfiability Solvers (SAT). The SMT has been used in many fields including deductive software verification. Moreover, recent applications of computer science including planning, model checking, and automated test generation finding, also consider SMT as an important verification tool [14]. (Readers are encourage to read [11] for the use of SMT-Lib in the verification of OSPF routing protocol.) Multiple solvers are available that support SMT-LIB, such as Beaver, Boolector, CVC4, MathSAT5, Z3, and OpenSMT. We used Z3 [14] solver in our study, which is a high performance theorem prover developed at Microsoft Research.

4 MODELING THERMAL CHARACTERISTICS OF CYBER PHYSICAL DC

We model DC as a CPS, where the logical classification is made between the computational section and supporting infrastructure. The computational section, such as scheduling, that participates in the distribution, processing, and flow of tasks constitutes the *Cyber* portion. The supporting infrastructure, such as servers, and switches, constitutes the *Physical* portion. The cyber portion performs computations or any other task to deliver the specified QoS attributes. In return, the physical portion emits thermal energy into the DC environment that raises the temperature. In this paper, we present a methodology that analyzes the thermal characteristics of cyber and physical portions in a unified way, to maintain a specified range of thermal threshold in the CPS. Generally, there are three main contribu-

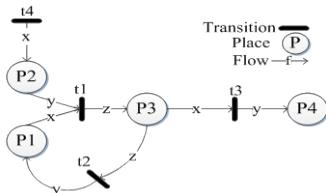


Fig. 1. An example High-Level Petri Net.

tors in the power consumption of a DC, namely: **(a)** servers (40-55%), **(b)** Data Center Networks (DCN) (10-25%), and **(c)** cooling systems, such as CRAC (15-30%). We perform the thermal modeling and analysis of servers and DCNs only. The CRAC units are reactive systems, where the supplied temperature coming from CRAC depends on the overall temperature of DC environment. The proposed thermal aware strategy (Section 5) aims at maintaining unified thermal temperatures within the pods of the DC that will ultimately reduce the overall temperature of the DC. Therefore, by reducing the temperature of the DC we are indirectly controlling the CRAC supplied temperature, which is derived by the ambient DC temperature. It is noteworthy, that we are only interested in the modeling of thermal properties of the DC and not the performance. The DC is logically classified as the combination of the cyber and physical portion: $DC = DC(\text{Cyber}) + DC(\text{Physical})$.

The CPS is comprised of computing resources, such as servers and the network infrastructure, such as switches, interconnecting all of the computing resources (Fig. 2). The CPS follows a hierarchical model, where the computing resources reside at the lowest layer as depicted in Fig. 2. The network infrastructure can be considered as a multilayer graph [23]. The servers, access switches, and aggregate switches are assembled in modules (referred to as pod) and are arranged in three layers, namely: **(a)** access, **(b)** aggregate, and **(c)** server layer. The core layer is used to connect all of the independent pods together. Note that, the cyber portion resides within the physical portion. Therefore, we model DC in a unified way that can accommodate both, the cyber and physical section.

We divided the CPS model into two logical sections, namely, **(a)** Pods (zones) and **(b)** Core Layer Switches, as below:

$$DC = Pod_{\forall i \in k}(i) \cup C_{\forall q \in r}(q), \quad (1)$$

where $C(q)$ is the set of core layer switches and r is the total number of core switches (γ) in the network. $Pod(i)$ is the set of pods and k is the total number of pods in the network. Each access layer switch (α) is connected to n number of servers (S) in a pod. Moreover, every α is connected to every aggregate switch (δ) in the pod. The number of nodes (including S , α , and δ) in $Pod(i)$ can be calculated as:

$$Pod(i) = S_{(n \times m)}^i \cup \alpha_m^i \cup \delta_w^i, \quad (2)$$

where $S_{(n \times m)}^i$ represents a set of servers connected to α in Pod (i). The α_m^i represents access layer switches in Pod (i), where m is the total number of α in Pod (i). The δ_w^i represents aggregate layer switches and w is the number of δ in Pod (i). The components in CPS work individually

or cooperatively to accomplish the assigned tasks. According to the law of energy conservation, energy can neither be created nor destroyed but it can be converted from one form to another. The mechanical energy is consumed by the physical portion when they perform cyber tasks and almost all the power drawn by the computing devices are dissipated as heat. We model the heat dissipation of every component within

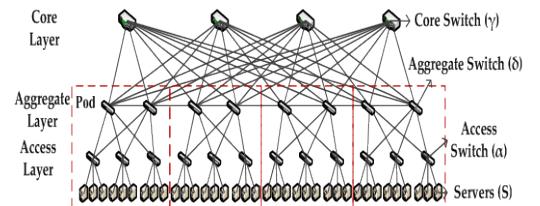


Fig. 2. Three-tier DC Architecture.

the pod, such as S , α , and δ . The heat dissipated by the S is represented as ζ_s and can be calculated as follows:

$$\zeta_s^{i,\alpha} = \zeta_0^{i,\alpha} + \zeta_p^{i,\alpha} + \zeta_m^{i,\alpha}, \quad (3)$$

where

$$\zeta_p^{i,\alpha} = \zeta_{rw}^{i,\alpha} + \zeta_{op}^{i,\alpha}, \quad (4)$$

The $\zeta_0^{i,\alpha}$ represents the heat dissipated as a result of the static power to keep the server awake, and $\zeta_p^{i,\alpha}$ represents the heat dissipation when the processing is being performed. The $\zeta_0^{i,\alpha}$ is fixed that does not change and is independent. However, $\zeta_p^{i,\alpha}$ is dynamic and is dependent on the workload. The $\zeta_m^{i,\alpha}$ represents the heat dissipated by the memory that includes energy consumed during the memory refresh operations. The $\zeta_p^{i,\alpha}$ is further decomposed into $\zeta_{rw}^{i,\alpha}$ that represents the heat dissipation because of the read and write operations, and $\zeta_{op}^{i,\alpha}$ is the heat dissipation as a result of the processing performed. We model switches as normal and high-end switches. The switches used in the core layer are usually high-end switches and dissipate more heat as compared to normal switches. We assume α and δ are normal switches and γ are high-end switches. The heat dissipated by the normal switches, such as α and δ is represented as ξ_n and can be calculated as:

$$\xi_n^i = (\xi_0 + \xi_f + \xi_b + \xi_p)^i, \quad (5)$$

where,

$$\xi_b = \xi_{ig} + \xi_e, \quad (6)$$

and

$$\xi_p = \xi_{p'} + \xi_{pr} + \xi_{rw}. \quad (7)$$

The ξ_0 represents the heat dissipation of the switch as a result of static power consumption, ξ_f represents the heat dissipation of the communication fabric used in the switch, ξ_b represents the heat dissipation of the buffer that includes ξ_{ig} and ξ_e , representing the heat dissipation of ingress and egress processing unit, respectively. The ξ_p represents the heat dissipation during the processing that includes $\xi_{p'}$ and ξ_{rw} , representing the static heat dissipation of switch processor, and when read and write operations are performed, respectively. The ξ_{pr} represents the heat dissipation due to the processing performed by the switch. The $\xi_{p'}$ and ξ_0 are constant. However, the ξ_p and ξ_b are dynamic and depend on the workload of the switch. The γ has different characteristics from α and δ . The α facilitates the connection of the network with the end node devices and for this reason it supports features, such as port security and VLANs. The δ manages or segments the traffic from the leaf nodes into VLANs and provide the information to the core layer. For the said reason, δ provides inter-VLANs routing functions to communicate. The γ are the high speed backbone of the network, so they have a very high forwarding rate. Moreover, they have the capability to support link aggregation to ensure adequate bandwidth and traffic routing coming from δ . Furthermore, γ have additional hardware redundancy features, such as redundant power supplies, to swap while the switch continues to operate. Because of the high workload carried out by γ , they dissipate more heat than α and δ . We, represent the heat dissipation of high-end switches (core layer) as \mathcal{K}_γ , which can be calculated using (5), (6), and (7). However, because of the workload and hardware redundancy the value of \mathcal{K}_γ must always be greater than ξ_n . In the previous discussion, we have modeled the heat dissipation of the individual nodes, as in (3)

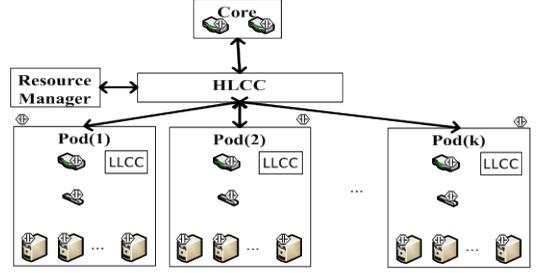


Fig. 4. HLCC and LLCC in DC.

and (5), involved in the CPS. The heat dissipated by all the servers in $Pod(i)$, represented as \mathcal{S}_s^i , can be calculated as:

$$\mathcal{S}_s^i = \sum_{p=1}^m \sum_{x=1}^n \zeta_x^{i,p}, \quad (8)$$

where the $\zeta_x^{i,p}$ represents the heat dissipation of S_x connected to m in $Pod(i)$. Moreover, the heat dissipation of all the α and δ in $Pod(i)$, represented as \mathcal{S}_δ^i and \mathcal{S}_γ^i , respectively, can be calculated as:

$$\mathcal{S}_\delta^i = \sum_{x=1}^m \xi_x^i, \quad (9)$$

$$\mathcal{S}_\gamma^i = \sum_{h=1}^w \xi_h^i, \quad (10)$$

where ξ_x^i and ξ_h^i represents the heat dissipated by access and aggregate switches in $Pod(i)$. Similarly, the overall heat dissipated by the CPS, represented as ψ_c , can be calculated as:

$$\psi_c = \sum_{i=1}^k (\mathcal{S}_s^i + \mathcal{S}_\delta^i + \mathcal{S}_\gamma^i) + \sum_{j=1}^r (\mathcal{K}_\gamma^j). \quad (11)$$

It is noteworthy, that the heat calculations performed at this point, do not consider the ambient effect involved in the CPS environment. The next paragraphs will discuss the process of ambient temperature and its effect on the heat dissipation of an individual component. The ambient temperature is the surrounding temperature. The Fig. 3 illustrates the effect of ambient temperature in the CPS environment. The red and blue lines in Fig. 3 depict the movement of hot and cold air, respectively. The hot air is exchanged amongst the racks, while the cooling is provided from the cooling devices, such as CRAC. Suppose there are \aleph number of nodes that participate in the heat dissipation of CPS. Two temperatures are associated with each node, the (a) input temperature (τ_{in}^i) and (b) output temperature (τ_{out}^i). The τ_{in}^i represents the input ambient temperature of node that includes the heat received from other thermal nodes. As depicted in Fig. 3, the τ_{in}^i of s_i involves the recirculation (red dotted lines) of hot air from other nodes and cooling temperature (τ_{sup}) from CRAC (more details on CRAC modeling can be seen in [20]). The heat dissipated by any node $i \in \aleph$ will change the τ_{out}^i . The τ_{in}^i and τ_{out}^i represent the temperature of the surroundings and not the node. However, the heat dissipated by the node (π_{out}^i) can affect the values of τ_{in}^i and τ_{out}^i . The input temperature of a node (π_{in}^i) can be calculated as:

$$\pi_{in}^i = \varrho^i(\tau_{in}^i), \quad (12)$$

where

$$\tau_{in}^i = \sum_{j=1}^{\aleph} (\pi_{out}^j) + \tau_{sup}. \quad (13)$$

The ϱ is an air coefficient that represents the product of air density (which changes from 1.205kg/m^3 at 20°C to 1.067kg/m^3 at 60°C), heat of air, and flow rate of air. The π_{out}^i can be calculated as:

$$\pi_{out}^i = \pi_{in}^i + \mathfrak{Q}^i, \quad (14)$$

where

$$\mathfrak{Q}^i = \varrho^i(\tau_{out}^i - \tau_{in}^i - \omega^*). \quad (15)$$

The \mathfrak{Q}^i represents the heat dissipation of a node $i \in \mathfrak{N}$ in proportion to the power consumed during the processing. The ω^* can be replaced by any of the heat dissipation value of three nodes. For instance, if the calculating node is γ , then ω^* can be replaced with \mathfrak{K} . Suppose we have the current power distribution of all the servers in $Pod(i)$, represented as a vector \vec{P}_i . The temperature profile of all the servers, represented as a vector \vec{T}_i , can be calculated based on the given power distribution. The current temperature of S_i in $Pod(j)$ is denoted as $t_{cur}^{i,j}$, which can be calculated as, $t_{cur}^{i,j} = \pi_{in}^i + \Delta t(c_i)$, where $\Delta t(c_i)$ represents the anticipated change in the temperature cause by executing a task c_i on S . According to the abstract heat model of DC, as discussed in previous works [27], the heat distribution and its effect on the surrounding machines can be represented as cross interference coefficient matrix. We follow the same model and compute the heat distribution of the servers using a matrix, represented as $h_{n \times n} = \{\partial_{i,j}\}$, which denotes the thermal effect of S_i on S_j and can be populated as:

$$\partial_{i,j} = \tau_{out}^i \times kt \times \frac{1}{\hat{h}_{ij}},$$

where kt is the thermal conductivity constant of the air and \hat{h} is the hop count of S_j from S_i .

5 THERMAL AWARE CONTROL STRATEGY (TACS)

We propose a thermal aware scheduling approach that uses High Level Centralized Controller (HLCC) and Low Level Centralized Controller (LLCC) to manage and control the thermal properties of CPS at different levels, such as: (a) low (server) level, (b) high (access and aggregate switch) level, (c) intra-pod level, and (d) inter-pod level. The goal is to eliminate hotspots and to maintain a uniform range of thermal threshold in every pod. Whenever a new job (a job can have multiple tasks) arrives to the CPS, the tasks are allocated to the specified server based on the thermal signatures. The HLCC and LLCC are proposed that perform the task allocation, task migration, and traffic redirection, based on the thermal analysis of the node or layer. As depicted in Fig. 4, there is LLCC in every pod that has the thermal information of all S, α , and δ . Every node in the CPS is equipped with a heat sensor that measures the temperature and the temperature is updated periodically to the LLCC.

In low (server) level (Fig. 5), the $\zeta_s^{i,\alpha}$ for all the $S \in Pod(i)$ is measured and observed through sensors periodically. Whenever the value of $\zeta_s^j, \forall j \in n$ exceeds the maximum threshold

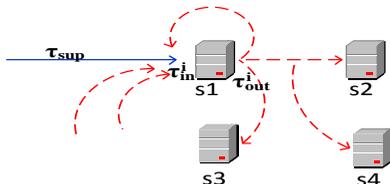


Fig. 3. Heat exchange amongst server nodes.

temperature of the server (τ_{max}^s), the LLCC migrates some tasks from S^j to S^l , where S^j and S^l are connected to the same α . For the tasks to be migrated successfully to S^l , the constraint $\zeta_s^l + \Delta T < \tau_{max}^s$, must be satisfied. The ΔT represents the anticipated increase in the temperature as a result of task migration. If the task migration is not possible amongst the servers under the α_i , then the servers belonging to $\alpha_j, \forall j \in m \wedge j \neq i$ are considered for the migration. The α_i and α_j belong to the same pod. Moreover, if there is no server available for the migration within the same pod, then inter-pod task migration is performed by enforcing the same constraint.

In high (access and aggregate) level (Fig. 6), the focus is to avoid the hotspot at access and aggregate layer of the CPS by redirecting the traffic from heavily loaded switches to the lighter ones. Redundant paths are available in the network infrastructure of DC that allows redirection of traffic from one switch to another (Fig. 2). The decisions for the redirections are made by LLCC considering the value of ξ_n for every switch. When the value ξ_n^i for α increases from τ_{max}^ξ , then task migration is performed by LLCC in the same way as performed in low level. The reason for the aforesaid is a fact that there is only one path between the access and the servers. However, in case of δ , redundant paths are available. Therefore, whenever the value of $\xi_n^i, \forall i \in w$, exceeds the maximum threshold temperature of the switch (τ_{max}^ξ), the LLCC instructs the lower level (server) to redirect the traffic from δ_i to δ_j where both δ belongs to the same pod. The redirection is allowed only if the $\xi_n^j + \Delta T < \tau_{max}^\xi$. If the redirection is not possible within the same pod, then inter-pod task migration is performed to take some load off from the switch.

The high level and low level are combined together to form an intra-pod control. The goal in intra-pod is to stabilize the temperature of the pod by maintaining the thermal signatures of server, access, and aggregate layer. Local decisions (within the same pod), such as task migration and redirection, are taken by LLCC to stabilize the temperature. However, the inter-pod migrations are performed with the consent of HLCC. Whenever, inter-pod actions have to be performed, the LLCC requests HLCC to provide information about other pods where the tasks can be migrated. Afterwards, the LLCC of the pods can communicate with each other to accomplish the task.

The inter-pod control is focused on maintaining the unified thermal threshold value in all the pods. The thermal signatures of nodes in CPS can evolve in order of minutes. Moreover, the power states of servers can change as frequent as milliseconds. Therefore, the threshold temperatures are not absolute values; rather it is a range within which the thermal signatures of the nodes and layers should lie. In inter-pod control, the HLCC

```

1:  for  $i \leftarrow 1$  to  $k$  do
2:       $\tau_\rho^i = \mathfrak{S}_s^i + \mathfrak{S}_\alpha^i + \mathfrak{S}_\delta^i$  // also use in inter-pod migration
3:  end for
4:  Select min ( $\tau_\rho^i$ )
5:  Get  $\zeta_s^{i,\alpha} \forall s \in n \wedge \alpha \in m$ 
6:  Select  $\zeta_s^{i,\alpha}$ , such that  $\zeta_s^{i,\alpha} < \zeta_y^{i,\alpha} \forall y \in n \wedge \alpha \in m \wedge y \neq s$ .
7:  Allocate  $c$  to  $\zeta_s^{i,\alpha}$ , iff  $\zeta_s^{i,\alpha} + \Delta t(c) < \tau_{max}^s$  //
8:  If  $\zeta_s^{i,\alpha} > \tau_{max}^s \forall s \in n \wedge \alpha \in m$ , then
9:      Migrate-task  $c$  from  $S_i$  to  $S_j$ , iff  $\zeta_j^{i,\alpha} + \Delta t(c) < \tau_{max}^s$ 
// intra-pod migration
10: end if

```

Fig. 5. Steps involved in low (server) level.

periodically monitors the average thermal values of each pod that it receives from sensors. Whenever the thermal signature of the $Pod(i)$ ($\tau_\rho^i = \mathcal{S}_s^i + \mathcal{S}_\delta^i + \mathcal{S}_g^i$) exceeds the maximum thermal threshold value of the pod (τ_{max}^ρ), the HLCC instructs the LLCC of $Pod(i)$ to migrate some tasks to $Pod(j), \forall j \in k \wedge j \neq i$. The migration can be successfully performed only if the $\tau_\rho^i + \Delta T < \tau_{max}^\rho$. Moreover, the server selection and task allocation performed in inter-pod control is same as in low level. The HLCC only has the coarse-grain information of the τ_ρ^i . The allocations of migrated tasks to servers are performed by LLCC through the use of fine-grained servers information.

All of the aforementioned controls work together to make sure that the CPS is operating under a specified temperature range. More detailed information, formal analysis, and behavior of the HLCC and LLCC will be discussed in the next section, using HLPN and Z language.

6 VERIFICATION USING HLPN, SMT-LIB, AND Z3 SOLVER

Verification is the process of demonstrating the correctness of an underlying system [17]. Two parameters are required to verify a model of a system: (a) specification and (b) properties. In this study, we use bounded model checking [5] technique to perform the verification, using SMT-Lib and Z3 solver. In bounded model checking, the description of any system is verified, whether any of the acceptable inputs drives the system into a state where the system always terminates after finite number of steps.

Definition 2 (Bounded Model Checking) [5]. Formally, given a Kripke Structure $M = (S, S_0, R, L)$ and a k bound, the bounded model checking problem is to find $\{M \models_k Ef\}$ where: S is the finite set of states, S_0 is the set of initial states, R is the set of transitions such that $R \subseteq S \times S$, and L is the set of labels. The bounded model checking problem is to find an execution path in M of at most length k that satisfies a formula f .

A path in a Kripke structure can be stated as an infinite sequence of states represented as $\rho = S_1, S_2, S_3, \dots$ such that for $\forall i \geq 0, (S_i, S_{i+1}) \in R$. The model M may produce a path set $= S_1, S_2, S_1, S_2, S_3, S_3, \dots$. To describe the property of a model some formal language, such as CTL*, CTL, or LTL is used. (Readers are encouraged to see [13] for more details about the CTL*.) For a model to be correct, the states must satisfy the formulas (Definition 2) under a specific bound.

Definition 3 (SMT Solver) [25]. Given a theory \mathcal{I} and a formula f , the SMT Solvers perform a check whether f satisfies \mathcal{I} or not.

```

1:  for  $i \leftarrow 1$  to  $k$  do
2:    Compute  $\xi_n^i \forall i \in m \wedge i \in w$ 
3:    If  $\exists \xi_n^i \in w$  such that  $\xi_n^i > \tau_{max}^\xi$ , then
4:      Redirect  $nt$  from  $\delta_i$  to  $\delta_j$ , iff  $\xi_n^j + \Delta t(nt) < \tau_{max}^\xi$ 
5:    end if
6:    If  $\exists \xi_n^i \in m$  such that  $\xi_n^i > \tau_{max}^\xi$ , then
7:      Migrate-task  $c$  from  $S_i$  to  $S_j$ , iff  $\xi_j^{i,\alpha} + \Delta t(c) <$ 
       $\tau_{max}^c \wedge \xi_n^j + \Delta t(nt) < \tau_{max}^\xi$  // intra-pod migration
8:    end if
9:  end for

```

Fig. 6. Steps involved in high (access and aggregate) level.

To perform the verification of the models using Z3 (an SMT Solver), we unroll the model M and the formula f that gives M_k and f_k , respectively. Moreover, the said parameters are then passed to Z3 to check if $M_k \models f_k$ [26]. The solver will perform the verification and provide the results as satisfiable (*sat*) or unsatisfiable (*unsat*). If the answer is *sat*, then the solver will generate a counter example, which depicts the violation of the property or formula f . Moreover, if the answer is *unsat*, then formula or the property f holds in M up to the bound k (in our case k is exec. time).

TABLE I

DATA TYPES USED IN THE HLCC AND LLCC MODEL

Types	Description
<i>Task</i>	A type for the representation of job.
<i>Res-Mat</i>	amount and type of resources available servers.
<i>Th_S</i>	a type for the thermal signature (Th. Sig) of the server.
<i>Th_P</i>	a type for the Th. Sig of the Pod.
<i>Th_Ac</i>	a type for the Th. Sig of the Access Switch.
<i>Th_Ag</i>	a type for the Th. Sig of the Aggregate Switch.
<i>Th_Co</i>	a type for the Th. Sig of the Core Switch.
<i>Res</i>	a type to represent the resources.
<i>RI</i>	a type to represent the Routing Information.
<i>Max_Th_P</i>	Max. Thermal Threshold (Th. Td) value of the Pod.
<i>Max_Th_S</i>	Max. Th. Td value of the Server.
<i>Max_Th_Ac</i>	Max. Th. Td of Access Switch.
<i>Max_Th_Ag</i>	Max. Th. Td value of Aggregate Switch.
<i>Max_Th_Co</i>	Maximum Thermal Threshold value of the core Switch.
Δt	Expected thermal dissipation of new task.

TABLE II

PLACES USED IN THE MODEL OF HLCC AND LLCC

Places	Mappings
$\varphi(job)$	$\mathbb{P}(Task \times Res)$
$\varphi(RM)$	$\mathbb{P}(Task \times Res-Mat \times Th_P \times Th_S)$
$\varphi(HL-CC)$	$\mathbb{P}(Th_P)$
$\varphi(Pod-Sen)$	$\mathbb{P}(Th_P)$
$\varphi(LL-CC)$	$\mathbb{P}(Th_S \times Th_Ac \times Th_Ag)$
$\varphi(AcS-S)$	$\mathbb{P}(Th_Ac)$
$\varphi(AgS-S)$	$\mathbb{P}(Th_Ag)$
$\varphi(CN-S)$	$\mathbb{P}(Th_S)$
$\varphi(Ags)$	$\mathbb{P}(RI)$
$\varphi(AcS)$	$\mathbb{P}(RI)$
$\varphi(CoS)$	$\mathbb{P}(RI)$
$\varphi(CoS-S)$	$\mathbb{P}(Th_Co)$
$\varphi(CNode)$	$\mathbb{P}(Task \times Res)$

6.1 Modeling HLCC and LLCC using HLPN

The HLPN model for HLCC and LLCC is shown in Fig. 7. The first step towards modeling using HLPN is to identify the required types, Places (P), and mapping (Definition 1). The types and the descriptions are shown in Table I and the mapping of P to types is depicted in Table II. The description and operation of the controllers are discussed in the previous section and now we can define formulas (pre and post-conditions) to map on transitions.

New tokens can only enter the model through the transition *New Jobs*. As seen in Fig. 7, no arc is incident on the aforementioned transition, which is why no pre-condition exists and the rules for the transitions can be written as: $R(New\ Jobs) = \exists j \in J \mid \bullet j = \emptyset$. Whenever the new jobs arrive, the resource manager checks if the resources required by the job are available or not. The said authentication is performed by the transitions *Job-Req-F* and *Job-Req-S*, mapped to the following formulas:

Moreover, inter-pod migration is also performed when the thermal signature of γ exceeds the specified maximum thermal threshold, as illustrated in (24).

7 RESULTS AND DISCUSSION

To demonstrate the effectiveness of our work in a real DC environment, we simulate the proposed strategies on a real data center workload obtained from the Center of Computational Research (CCR), State University of New York at Buffalo. All jobs submitted to the CCR are logged for a period of a month. The jobs and the logs from the CCR dataset are used as an input for our simulation of the proposed thermal aware strategy. The dataset had 22,700 jobs (127,000 tasks) recorded in one month of a time. The data center had 1056 distinct dual core servers. A server was based on the Dell 1056 PowerEdge SC1425 processor with 3.0 GHz speed, running x86-64 Linux operating system. The CCR data center was organized into 33 pods and each pod had 32 servers. Moreover, we also evaluate the proposed TACS by comparing with a classical First Come First Serve (FCFS), Genetic Algorithm (GA) based thermal aware scheduling [3], and Thermal Aware Task Allocation [18] approaches. We perform the comparison among the mentioned strategies based on the CCR dataset. Before going deeper into the details of the comparison, we first briefly discuss the existing approaches. The FCFS (sometimes referred as first-in, first-out) is possibly the most straightforward scheduling approach. The jobs are submitted to the scheduler, which dispatches the jobs based on the order of the jobs received. The approach in [3] follows the steps of GA. The first step is to construct a set of feasible solutions, which is the task allocation to the servers. Then, the selected solution is mutated (randomly interchange the task allocations within the solution) and mated (randomly select pairs of solution and exchange the subset of two task assignment to get two new solutions). The fitness function, which checks the highest inlet temperature of the selected assignment, is applied to all of the solutions that are formed as a result of mating and mutation, including the original solution. Finally, the solution having the lowest inlet temperature value from the set of highest inlet temperature values, obtained as a result of fitness function, is selected as a final solution. The last approach is TASA proposed in [18], which is based on the theory of coolest inlet that perform the assignment of hottest jobs to the coolest servers. The TASA algorithm sorts the servers in the increasing order of the temperatures. The jobs are sorted in a similar way but in the reverse order, such that the hottest job is first in the order. The hottest job is assigned to the coolest server and the thermal map of all the servers is updated.

The Fig. 8 depicts the average thermal signatures of the pods over the period of time, when the scheduling approaches are used. The epoch time stamp and average thermal signature of the pods at that particular time are plotted on x-axis and y-axis, respectively in Fig. 8. It can be observed from the Fig. 8 that the spread or the difference between the temperatures of the servers in the trend line of Fig. 8(a), (b), (c) is very wide at many time stamps. The aforesaid, identify the situation when the average temperature of some servers is lower than the rest of the servers in the DC. Particularly, at time stamps $1.2357E+9$, $1.2362E+9$, and $1.2372E+9$ in Fig. 8 (a), (b), (c), the thermal signatures of some pods are very low as compared to the rest, which shows the probable presence of the hotspots in DC.

The possible reason for the occurrence of the hotspots in Fig. 8(a) is the static assignment of tasks without considering the thermal status of the server. The aforesaid, possibly creates a scenario when higher task-temperature profiled jobs are assigned to the servers with high thermal signatures and low thermal impact jobs are assigned to low thermal signature servers. In such a scenario, the thermal signatures of the “hot” servers will increase, causing thermal imbalance among the servers and pods.

In Fig. 8(b), the reason for the imbalance thermal signatures is the random nature of the GA based approach. The selection of the feasible solution, the mutation, and the mating process, all are based on randomization. If the same set of pods and servers are selected in the solutions most of the time, then the fitness function performed on the selected solution will not provide any important information that will avoid the occurrence of the hotspots. Similarly, there is also a possibility that the number of tasks allocated to few pods and servers are relatively low as compared to the rest of the pods and servers in the DC. The aforementioned possibilities will allow some servers to have high thermal signatures while others have low thermal signatures, which will ultimately cause the hotspot in the DC. In Fig. 8(c), the thermal differences are lower than (a) and (b). However, there are still some time stamps, where some pods have high thermal signatures and some have low. The reason for the aforesaid is that the high thermal profile tasks are allocated to the coolest servers regardless of the overall thermal temperature of the pod and the recirculation effect that can cause the hotspots. The aforesaid can cause a situation, where the temperature of the server is low but the overall temperature of the pod in which the server lie is high. In such situation, the overall temperature of the pod increases that can possibly create a hotspot. In TACS, as shown in Fig. 8(d), the differences of the temperatures amongst the servers are low and there are no hotspots. As stated in Sections 5 and 6, the selection of the pods and servers to allocate the task is based on the thermal signatures. Moreover, the HLCC and LLCC periodically monitor the thermal signatures of the pods and servers, and perform task migration or redirection to maintain unified range of temperatures in the pods. Therefore, the trend of thermal signatures followed in Fig. 8(d) is more congested and unified as compared to the trend followed in rest of the approaches. We plot the average difference between the hottest and coolest servers over the period of time (as shown in Fig. 9). The larger and more frequent the differences are, the higher the thermal imbalance will be. We can see that the differences in TACS (d in Fig. 9) are very low and less frequent as compared to the other approaches that indicate the thermal balance achieved by using TACS. However, the other approaches have high differences and are occurring frequently, which indicates the thermal imbalance and occurrence of the hotspots.

To verify, the HLPN models are first translated into SMT. Then, the models along with the properties are provided to the Z3 solver, which checks if the properties are satisfied by the models or not. It is noteworthy, that the goal of the verification is to demonstrate the correctness of the models, based on the desirable properties, such as the presence of the hotspots. The results in Fig. 10 depict the time taken by the Z3 solver to check the satisfiability of the models, based on the stated property. The property we verify is that, there must be no hotspots in the DC. The verification results reveal the absence of the hotspots when TACS is used.

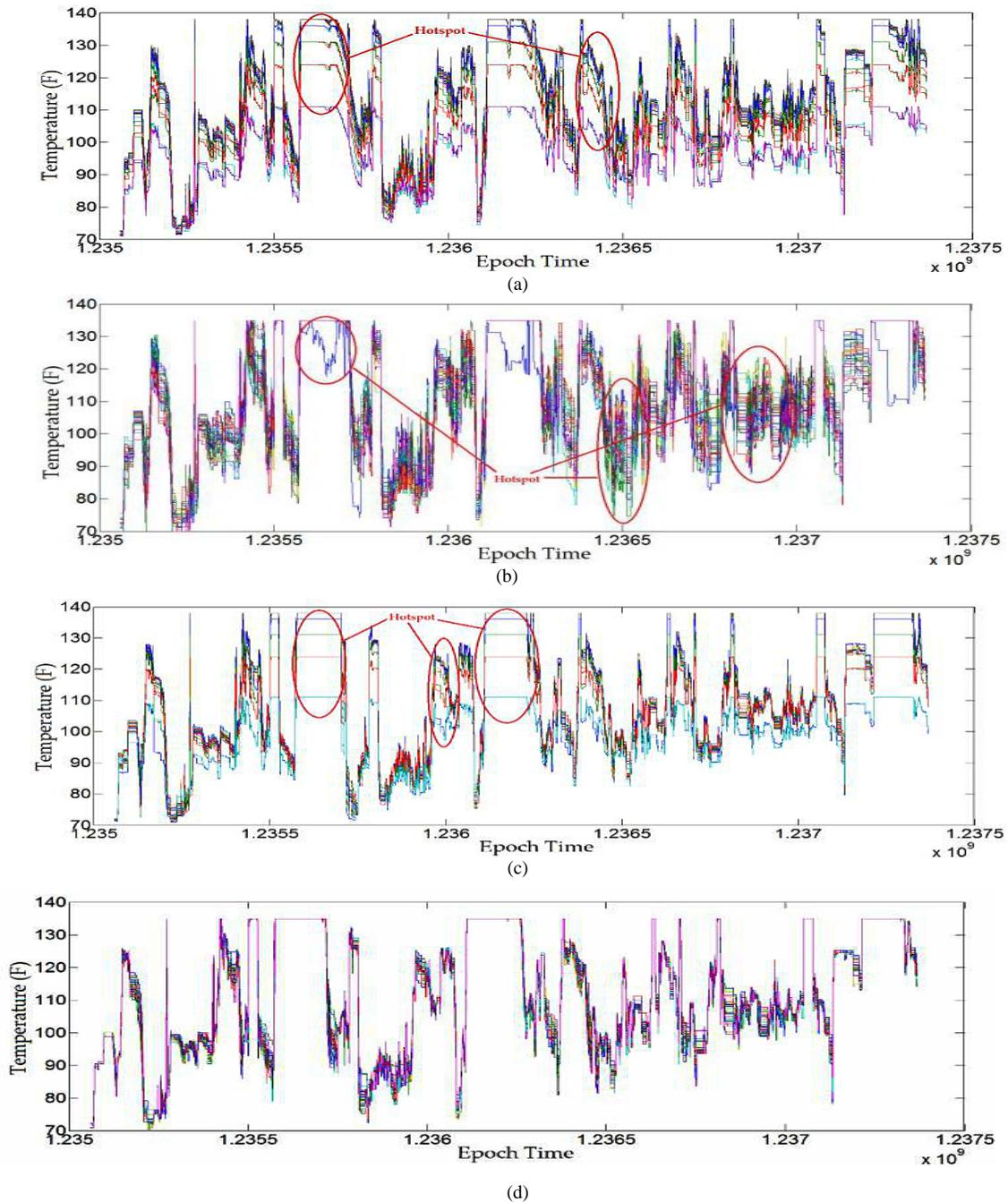


Fig. 8. Comparison of average thermal signatures of the pods using: (a) FCFS, (b) GA-based, (c) TASA, and d) TACS.

The execution time serves as a bound over the verification models. The simulation and verification results reveal that our strategy is consistent and provides better results as compared to the other scheduling approaches. We reduce the possibility of hotspots in our strategy through strategic decisions performed by HLCC and LLCC based on the thermal signatures of the components.

8 CONCLUSIONS

In this paper, we modeled DC as a CPS to capture the thermal evolution and properties presented by DC components. Moreover, we proposed a TACS that take strategic decisions to achieve thermal uniformity within a DC. A comparative analysis was performed, which reveals the effectiveness of our strategy. Furthermore, to demonstrate the correctness of our ap-

proach we performed formal analysis, modeling, and verification using HLPN, SMT-Lib, and Z3 solver. The automated verification performed using SMT-Lib and Z3 solver authenticate the correctness of our approach as compared to other approaches, where hotspots were identified. In future, we will analyze the effect of workload migration on network throughput and latency. Moreover, the effect of thermal balancing towards attaining efficient power consumption will also be performed.

References

- [1] R. Buyya, S.Y. Chee, and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities," *IEEE HPCC*, pp. 5-13, 2008.
- [2] S. U. R. Malik, S. U. Khan, and S. K. Srinivasan, "Modeling and Analysis of State-of-the-art VM-based Cloud Management Platforms," *IEEE Transactions on Cloud Computing*, pp. 50-63, 2013.

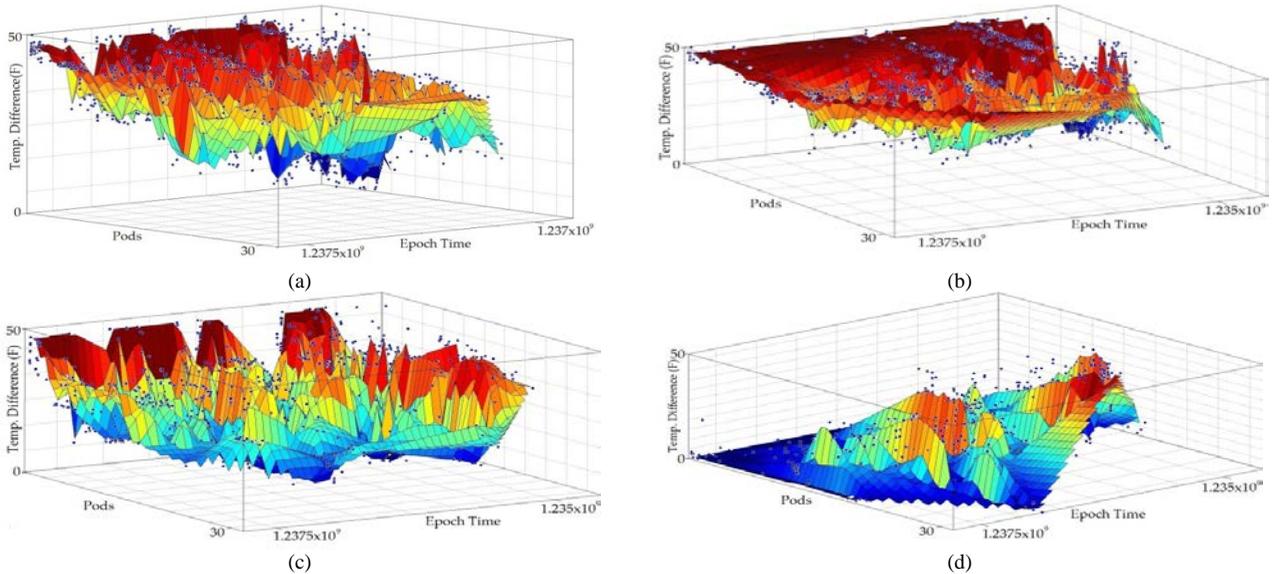


Fig. 9. Comparison of average thermal signature difference between the highest and lowest servers: (a) FCFS, (b) GA-based, (c) TASA, and (d) TACS.

- [3] Q. Tang *et al.*, "Energy-efficient thermal aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach," *IEEE TPDS*, vol. 19, no. 11, 2008, pp. 1458–1472.
- [4] J. Koomey, Growth in data center electricity use 2005 to 2010, CA: Analytics Press. July, <http://www.analyticspress.com/datacenters.html>.
- [5] A. Biere, A. Cimatti, E. Clarke, O. Strichman, and Y. Zhu, "Bounded model checking," *Advances in Computers*, vol. 58, 2003.
- [6] J. Shuja, S. A. Madani, K. Bilal, K. Hayat, S. U. Khan, and S. Sarwar, "Energy-Efficient Data Centers," *Computing*, vol. 94, no. 12, 2012.
- [7] J. Moore, J. Chase, P. Ranganathan, and R. Sharma, "Making scheduling "cool": temperature-aware workload placement in data centers," *In USENIX*, pp. 61–75, 2005.
- [8] L. Ramos and R. Bianchini, "C-oracle: predictive thermal management for data centers," *HPCA*, pp. 111–122, 2008.
- [9] L. Parolini, B. Sinopoli, B. Krogh, and W. Zhikui, "A Cyber-Physical Systems Approach to Data Center Modeling and Control for Energy Efficiency," *Proceedings of IEEE*, vol. 100, no. 1, pp. 254–268.
- [10] K. D. Kim and P. R. Kumar, "Cyber-physical systems: A perspective at the centennial," *Proceedings of IEEE*, vol. 100, pp. 1287–1308, 2012.
- [11] S. U. R. Malik, S. K. Srinivasan, S. U. Khan, and L. Wang, "A Methodology for OSPF Routing Protocol Verification," *Conference on Scalable Computing and Communications*, Dec. 2012.
- [12] H. Aydin and D. Zhu, "Reliability-aware energy management for periodic real-time tasks," *IEEE TC*, vol. 58, no. 10, 2009, pp. 1382–1397.
- [13] M. Maidl, "The common fragment of CTL and LTL," *Symposium on Foundations of Computer Science*, pp. 643–652, 2000.
- [14] SMT-Lib <http://smtlib.cs.uiowa.edu/>, accessed Jan. 2013.
- [15] J. Leverich, M. Monchiero, V. Talwar, P. Ranganathan, and C. Kozyrakis, "Power management of datacenter workloads using per-core power gating," *Computer Architecture Letters*, 2009, vol. 8, pp. 48–51.
- [16] Z. Jian-Hui and Y. Chun-Xin, "Design and simulation of the cpu fan and heat sinks," *IEEE Transactions on Components and Packaging Technologies*, vol. 31, no. 4, 2008, pp. 890–903.
- [17] S. Nakajima, "Model-checking Verification for Reliable Web Service," *OOWS*, 2002.
- [18] L. Wang *et al.*, "Towards thermal aware workload scheduling in a data center," *I-SPAN*, pp. 116–122, 2009.
- [19] J. Moore, J. Chase, and P. Ranganathan, "Weatherman: Automated, online and predictive thermal mapping and management for data centers," *IEEE ICAC*, pp. 155–164, 2006.
- [20] N. Kumari, Z. Wang, C. Bash, R. Shih, T. Cader, and C. Felix, "Cooling capacity and cost metrics for energy efficient workload placement in datacenters," *In IEEE ITherm*, pp. 799–805, 2012.
- [21] C. Bash, C. Patel, and R. Sharma, "Dynamic thermal management of air cooled data centers," *IEEE ITherm*, pp. 445–452, 2006.
- [22] G. Varsamopoulos, M. Jonas, J. Ferguson, J. Banerjee, and S. Gupta, "Using transient thermal models to predict cyber physical phenomena in data centers," *Sustainable Computing: Informatics and Systems*, 2013, vol. 3, no. 3, pp. 132–147.
- [23] K. Bilal, M. Manzano, S. U. Khan, E. Calle, K. Li, and A. Y. Zomaya, "On the Characterization of the Structural Robustness of Data Center Networks," *IEEE TCC*, vol. 1, no. 1, pp. 64–77, 2013.
- [24] T. Murata, "Petri Nets: Properties, Analysis and Applications," *Proc. IEEE*, vol. 77, no. 4, 1989, pp. 541–580.
- [25] L. Cordeiro, B. Fischer, and J. Marques-Silva, "SMT-based bounded model checking for embedded ANSI-C software," *ASE*, 2009.
- [26] M. K. Ganai and A. Gupta, "Accelerating high-level bounded model checking," *in ICCAD*, 2006, pp. 794–801.
- [27] Q. Tang, T. Mukherjee, S.K.S. Gupta, and P. Cayton, "Sensor-Based Fast Thermal Evaluation Model for Energy Efficient High-Performance Datacenters," *ICISIP*, Dec. 2006.
- [28] J. Wan *et al.*, "From machine-to-machine communications towards cyber-physical systems," *Computer Science and Information Systems*, vol. 10, no. 3, 2013, pp. 1105–1128.

Saif U. R. Malik received a PhD from North Dakota State University, USA. His research interests include formal verification, modeling, cyber physical systems, and large scale computing systems.

Kashif Bilal received a PhD from North Dakota State University. His research interests include data center networks, distributed computing, and energy efficiency.

Samee U. Khan Samee U. Khan is Associate Professor at the North Dakota State University, USA. Prof. Khan's research focuses on the optimization, robustness, and security of systems. His work has appeared in over 300 publications. For more information, visit: <http://sameekhan.org/>.

Bharadwaj Veeravalli received a PhD degree from the Department of Aerospace Engineering, IIS, Bangalore. His research interests are cloud/grid/cluster computing, bioinformatics and computational biology, and multimedia computing.

Keqin Li is a SUNY distinguished professor of computer science in the State University of New York at New Paltz. His research interests are design and analysis of algorithms, parallel and distributed computing, and computer networking. He has published over 280 journal articles.

Albert Y. Zomaya is the Chair Professor of High Performance Computing & Networking in the School of IT, Sydney University. He is the author/co-author of 07 books, more than 540 papers, and the editor of 16 books and 23 conference proceedings. He is a Fellow of AAAS, IEEE and IET (UK).

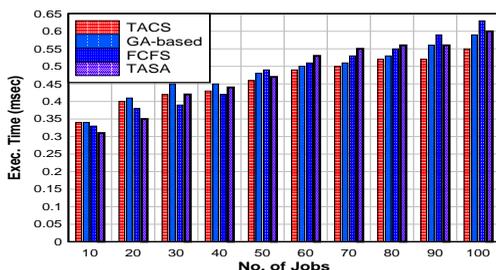


Fig. 10. Verification time comparison of the approaches.